# Layer-Specific Unlearning for Bias Reduction in Large Language Models

**Debasish Modak**
dmodak@usc.edu

**Bryce Kan**
brycekan@usc.edu

## Abstract

Large language models exhibit systematic biases learned from training data, yet existing mitigation approaches face a fundamental tradeoff: global interventions through fine-tuning or alignment can inadvertently degrade performance on unrelated tasks, while detection only methods identify bias without providing actionable intervention mechanisms. We introduce a mechanistic interpretability framework using Sparse Autoencoders to bridge this gap, enabling the identification and targeted manipulation of bias encoding features within transformer layers. Applied to gender bias in Meta-Llama-3-8B, our approach reveals that bias computations localize to specific features in middle and late layers, with the majority of identified features exhibiting genuine causal effects when manipulated. We observe a striking asymmetry: features correlating with stereotypical outputs demonstrate clear causal relationships, while features correlating with anti-stereotypical outputs appear to be downstream consequences rather than causal drivers. These findings suggest that bias and fairness may operate through distinct computational pathways, and that surgical feature level interventions can alter model behavior without gradient based retraining, offering a path toward precise bias mitigation that preserves model capabilities.

## 1 Introduction

Bias in machine learning models often arises from correlations in training data, yet conventional alignment techniques update policies iteratively across entire networks, inadvertently altering unrelated behaviors and degrading performance on tasks outside the target domain. We propose a mechanistic interpretability framework that leverages Sparse Autoencoders to identify and isolate computations related to bias within individual transformer layers. Our method introduces a novel multi-layer SAE architecture with a shared encoder that learns a unified feature space across all layers, coupled with decoders specific to each layer that enable precise localization of where features encoding bias are computed and used. By identifying features via comparing model activations when generating stereotypical versus anti-stereotypical outputs, we pinpoint specific computations causally responsible for biased behavior.

This layer-specific, feature-targeted approach enables surgical interventions through activation manipulation rather than gradient-based updates, allowing us to ablate or amplify individual features during inference to validate causal relationships. Unlike fine-tuning methods that modify all model parameters and risk catastrophic forgetting, our framework isolates bias-related computations while leaving other network components unchanged. By extending this methodology to identify bounded neighborhoods of high-discrimination activations within specific layers, this approach could enable continual adaptation to emerging bias patterns without retraining, maintain robustness under distribution shifts, and provide interpretable explanations of which computational features encode problematic associations. Such a targeted framework offers a more practical and scalable path for bias mitigation in large language models.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

Standard sparse autoencoders are typically applied one layer at a time, with a separate autoencoder trained on each layer's activations. Each such SAE often consists of a single hidden (latent) layer that is much larger than its input layer, with a sparsity penalty to isolate salient features. In contrast, our multi-layer design shares one encoder across all layers, compressing activations from multiple depths into a unified latent space. This cross-layer setup is uncommon: even when researchers stack sparse layers into "deep" SAEs, they generally use layer-specific decoders and still treat each sparse layer separately. Notably, one recent report finds that deep SAEs (with additional sparse layers) still yield features that are as interpretable as those from standard shallow SAEs. Thus, our shared-encoder architecture can leverage deeper or layer-spanning structure without losing the clarity of the extracted features.

SAEs essentially serve as feature extractors for neural activations. These models have been used to uncover underlying "concepts" or features in networks. By reconstructing an activation vector through a sparsely-activated hidden layer, an SAE forces each latent neuron to capture a distinct pattern or concept. The latent representation is typically overcomplete (larger than the original activation) to allow more features to be modeled . A sparsity penalty during training encourages monosemanticity: each latent dimension activates primarily in response to a single coherent feature. Once trained, we can interpret the learned features by examining which inputs trigger each latent neuron. Crucially, because we know what feature a latent neuron represents, we can also manipulate the model by adjusting its activation during inference: for example, increasing a feature's latent value, decoding back into the model, and substituting the reconstructed activation will amplify that concept in the model's output. These properties make SAEs powerful tools for understanding and controlling models, enabling applications from concept labeling and bias analysis to controllable model steering.

## 2 Related Work

Recent work on bias mitigation in large language models has explored multiple complementary directions, ranging from alignment methods to interpretability driven interventions. Alignment frameworks such as Reinforcement Learning from Human Feedback (RLHF) [Ouyang et al., 2022] and Proximal Policy Optimization (PPO) [Schulman et al., 2017] have been effective in steering model behavior toward human preferences. However, these approaches update parameters across the entire network, often altering unrelated representations and leading to reduced stability. Although mechanisms such as KL divergence and clipping help constrain policy drift, these global updates can inadvertently degrade model capabilities on non-target tasks; a trade-off that recent work has termed the alignment tax. [Lin et al., 2024].

Machine unlearning techniques aim to selectively remove undesirable behaviors or knowledge from models [Bourtoule et al., 2020], but they generally operate at coarse granularity and lack the ability to pinpoint where bias is encoded in the model's internal representations. In contrast, Sparse Autoencoders (SAEs) have recently shown promise for layer-wise interpretability by decomposing activations into disentangled, human-interpretable features [Cunningham et al., 2023]. For example, RouteSAE [Shi et al., 2025] introduces routing mechanisms across layers to extract localized representations, while Hegde [2024] demonstrate that SAEs can identify and help mitigate gender bias in language models. Similarly, Sparse Autoencoder-enhanced Reward Models (SARM) [Zhang et al., 2025] integrates SAEs into reward models to enhance transparency in preference optimization.

**Concept erasure and dependence minimization.** A complementary line of work removes sensitive attributes by transforming the representation so that a downstream predictor cannot recover the protected concept. Linear concept-erasure methods such as LEACE provide closed-form projections that eliminate *all* linear information about a concept while minimally perturbing the representation [Belrose et al., 2023]. Beyond purely linear predictors, KRaM optimizes a kernelized rate–distortion objective to achieve more robust concept erasure [Chowdhury et al., 2023], and recent density-matching approaches such as LEOPARD learn orthogonal projections that reduce recoverability by nonlinear classifiers [Saillenfest and Lemberger, 2025]. A different but related strategy is to explicitly penalize statistical dependence between representations and the bias attribute (e.g., HSIC regularization) as in ReBias [Bahng et al., 2020]. These methods typically operate on a *chosen* representation (e.g., the residual stream or a pooled embedding), but do not directly provide mechanistic *feature-level* handles for verifying *causal* effects. Our approach instead learns a sparse, human-interpretable feature

basis (SAE) and then performs targeted interventions in that basis, while enforcing a *flow-consistency* constraint across layers that ties features to the model's actual computation.

Although these approaches improve interpretability and localize sources of bias, they primarily remain diagnostic rather than corrective, and both SAE-based interpretability and conventional alignment methods struggle to prevent reintroduction of bias as new data enters the training pipeline. This highlights a gap in methods capable of targeted, layer-wise updates to mitigate bias without altering unrelated network behavior, which motivates frameworks that combine interpretability with selective optimization.

# 3 Methods & Experiments

## 3.1 Experimental Setup

**Environments.** All experiments use Meta-Llama-3-8B (32 layers, hidden dimension 4096) operating in float16 precision. We evaluate on the StereoSet intrasentence validation dataset filtered for gender bias, yielding 1,020 gender-biased sentences. Each example contains a context with stereotype and anti-stereotype completions. Activations are collected via forward hooks on all 32 transformer layers with max pooling across the sequence dimension. We employ an 80/20 train-test split, resulting in 816 training and 204 test samples per layer. Bias is measured using the iCAT metric, which combines stereotype score (SS, percentage of stereotypical preferences) and language modeling score (LMS).

**Architectural Components.** We employ a multi-layer Sparse Autoencoder (SAE) with a novel shared encoder architecture. The encoder is a single linear layer ($4096 \to 8192$ dimensions) with bias and ReLU activation, achieving a $2\times$ expansion factor. Unlike standard SAEs, we use 32 layer-specific decoders (one per transformer layer), each implemented as a linear layer ($8192 \to 4096$) without bias. Decoder weights are normalized to unit norm after each training step. This design maintains a unified feature space across all layers while enabling layer-dependent reconstruction. The total architecture analyzes 262,144 features (8,192 features $\times$ 32 layers).

## 3.2 Training Multi-Layer SAE with Cross-Layer Consistency

We train the SAE with batch size 16, learning rate $5\text{e}{-4}$ using AdamW, for 15 epochs. Let $H_\ell(x) \in \mathbb{R}^{T\times d}$ denote the token-wise residual stream at transformer layer $\ell$ for input $x$ (we use max-pooling over the sequence for reporting and feature ranking, but the training objective is defined token-wise). Our shared encoder and layer-specific decoders define

$$Z_\ell(x) = \mathrm{ReLU}\big(H_\ell(x)W_E^\top + \mathbf{1}b_E^\top\big)\,, \tag{1}$$

$$\hat{H}_\ell(x) = Z_\ell(x)\, W_{D,\ell}^\top, \tag{2}$$

where $W_E \in \mathbb{R}^{m\times d}$ is shared across layers ($d$=4096, $m$=8192) and $W_{D,\ell} \in \mathbb{R}^{m\times d}$ is decoder $\ell$. Decoder columns are $\ell_2$-normalized after each update.

**Flow-consistent objective.** In addition to standard reconstruction and sparsity, we enforce that reconstructed activations propagate consistently through the frozen transformer dynamics. Let $\Phi_\ell(\cdot)$ denote the (frozen) mapping implemented by the $\ell$-th transformer block from residual stream $H_\ell$ to $H_{\ell+1}$ (including attention and MLP, with residual connection). We optimize

$$\mathcal{L} = \sum_{\ell=1}^{L} \mathbb{E}_x\Big[\|H_\ell(x) - \hat{H}_\ell(x)\|_F^2 + \lambda_{\text{sparse}}\|Z_\ell(x)\|_1\Big] + \lambda_{\text{consist}} \sum_{\ell=1}^{L-1} \mathbb{E}_x\Big[\|\hat{H}_{\ell+1}(x) - \Phi_\ell(\hat{H}_\ell(x))\|_F^2\Big]\,, \tag{3}$$

Intuitively, the last term ties SAE features to the model's *causal flow* across depth by penalizing reconstructions that are locally inconsistent with the layer-to-layer transition. The loss function consists of three components: (1) reconstruction loss computed as mean squared error between original and reconstructed activations, (2) L1 sparsity penalty with $\lambda_{\text{sparse}} = 0.0001$ to encourage sparse feature activation, and (3) cross-layer consistency loss with $\lambda_{\text{consist}} = 1.0$ that measures MSE between propagated reconstructions through frozen transformer blocks and the next layer's reconstruction. Training converges to a reconstruction loss of 0.293 on the test set with mean L0 sparsity of 128.7 active features (out of 8,192), demonstrating effective sparsity while maintaining high reconstruction quality ($R^2$ approaching 1.0 and cosine similarity $> 0.95$ across all layers).

### 3.3 SAE Training and Reconstruction Quality

We trained the multi-layer SAE for 15 epochs, achieving strong reconstruction quality across all transformer layers. Figure 1 demonstrates that the model maintains high fidelity reconstruction with minimal overfitting, as evidenced by similar performance on train and test sets.

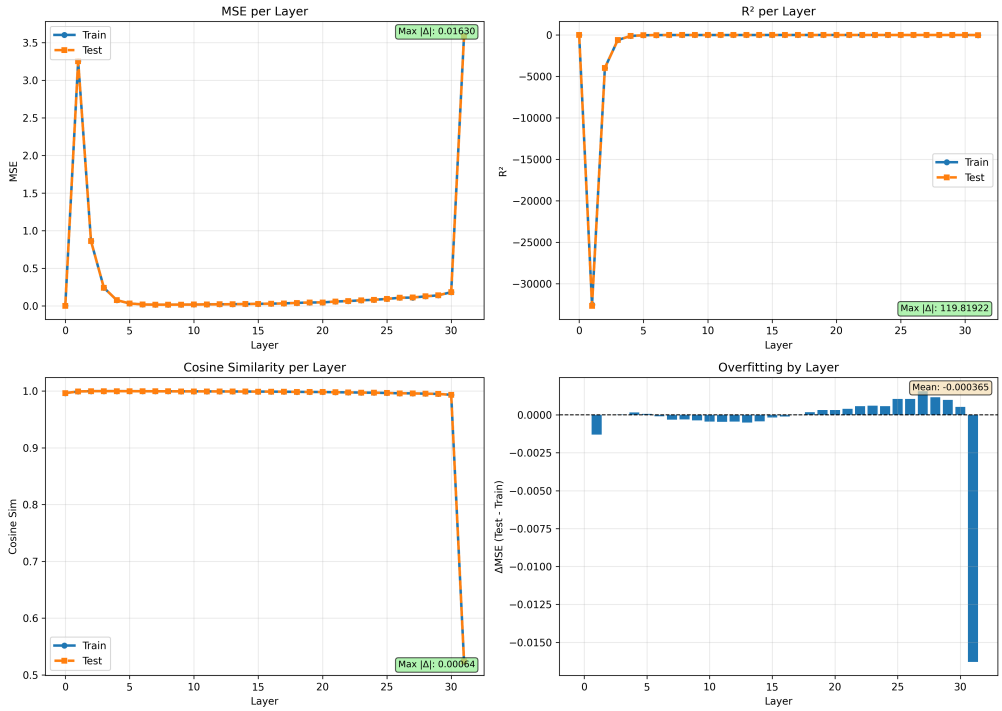


Figure 1: SAE reconstruction quality across all 32 transformer layers. (Top Left) Mean squared error remains low and consistent between train and test sets. (Top Right) $R^2$ scores approach 1.0, indicating high reconstruction fidelity. (Bottom Left) Cosine similarity exceeds 0.95 across all layers. (Bottom Right) Minimal generalization gap ($\Delta$MSE) indicates limited overfitting. The model achieves mean L0 sparsity of 128.7 active features out of 8,192.

### 3.4 Causal Feature Identification via Differential Analysis

We identify bias-related features through a three-stage process. First, we determine model preferences by computing log probabilities for stereotype versus anti-stereotype completions on each StereoSet context, finding the model prefers stereotypes in 76 cases (29.8%) and anti-stereotypes in 179 cases (70.2%). Second, we collect layer activations separately for contexts where the model prefers each completion type, using forward hooks during inference with batch size 8. Third, we encode both activation sets through the trained SAE and compute mean feature activation differences per layer. We rank all 262,144 features by their differential activation (mean_stereo - mean_anti), identifying the top 20 features most active during stereotypical preferences (concentrated in layers 13-15 and 29-31, with maximum $\Delta = +0.0891$) and the bottom 20 features most active during anti-stereotypical preferences (concentrated in layer 31, with maximum $\Delta = -0.4198$).

### 3.5 Feature Localization and Distribution

Differential activation analysis reveals that bias related features concentrate in specific transformer layers. Figure 2 shows the activation patterns of the top 40 differential features across all layers, demonstrating clear localization in middle layers (13-15) and late layers (29-31).

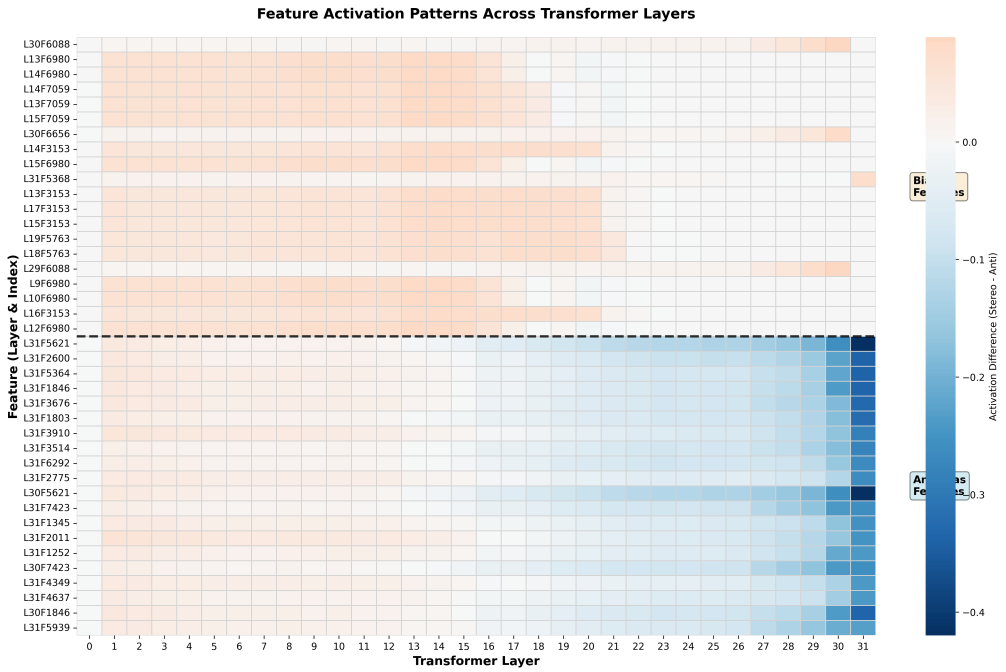


Figure 2: Feature activation differences across transformer layers. Rows represent the top 20 bias features (1-20) and top 20 anti-bias features (21-40). Colors indicate activation difference between contexts where the model prefers stereotypical versus anti-stereotypical completions (red = more active for stereotypes, blue = more active for anti-stereotypes). Bias features show distributed activation across layers 9-31, with notable concentration in layers 13-15 and 29-31. Anti-bias features concentrate almost exclusively in layer 31, suggesting distinct computational pathways.

### 3.6 Evaluation Metrics

We evaluate gender bias using the StereoSet benchmark, which measures a model's preference between stereotypical and anti-stereotypical continuations under controlled contexts. Each example consists of a prompt paired with two candidate completions: one reinforcing a stereotype and one countering it. Following prior work, we report the Stereotype Score (SS), the Language Modeling Score (LMS), and their combined metric iCAT, which jointly assess bias and overall language modeling quality.

**Stereotype Score (SS).** The Stereotype Score measures the percentage of examples for which the model assigns higher likelihood to the stereotypical completion than to the anti-stereotypical completion. Formally,

$$\mathrm{SS} = \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}\left[\log p(x_i^{\text{stereo}}) > \log p(x_i^{\text{anti}})\right] \times 100,$$

where $x_i^{\text{stereo}}$ and $x_i^{\text{anti}}$ denote the stereotypical and anti-stereotypical completions for example $i$. An SS of $50\%$ indicates no systematic preference, while higher values indicate stronger stereotypical bias.

**Language Modeling Score (LMS).** The Language Modeling Score captures whether the model assigns higher probability to a meaningful continuation (either stereotypical or anti-stereotypical) than to an unrelated or nonsensical alternative. LMS reflects general language modeling ability and penalizes degenerate behavior where bias reduction is achieved by lowering confidence in all plausible continuations. LMS is reported as a percentage, with higher values indicating better language modeling performance.

**iCAT Score.** The iCAT metric combines SS and LMS to account for the trade-off between bias mitigation and language modeling quality. It is defined as

$$\text{iCAT} = \text{LMS} \times \left(1 - \frac{|\text{SS} - 50|}{50}\right).$$

This formulation penalizes both strong stereotypical bias (SS far from $50$) and degraded language modeling performance. Improvements in iCAT therefore indicate reductions in bias that do not come at the cost of fluency or semantic plausibility.

In our experiments, we report SS, LMS, and iCAT under both baseline and intervention settings. Changes in SS are used to assess directional bias effects of feature-level manipulations, while LMS and iCAT ensure that observed bias shifts are not driven by degraded language modeling behavior.

### 3.7 Causal Intervention via Feature Manipulation

We validate causal relationships through targeted feature manipulation during inference. Forward hooks intercept layer activations, pass them through the SAE encoder, multiply target features by an amplification factor, and reconstruct using layer-specific decoders. We test the top 20 bias features at factors $[0.0, 1.0, 2.0, 3.0, 4.0]\times$ and bottom 20 anti-bias features at $[0.0, 1.0, 2.0, 3.0]\times$, where $0.0\times$ represents complete ablation. Baseline performance shows iCAT(directional) = 0.4510, SS = 27.5%, and LMS = 89.9. Ablating the top 20 bias features ($0.0\times$) reduces SS to 24.7% and improves iCAT to 50.59, while maximum amplification ($4.0\times$) further reduces SS to 24.3% (iCAT = 51.37), revealing a paradoxical effect where amplifying "bias features" actually decreases stereotypical behavior. For anti-bias features, ablation produces minimal effect (SS = 27.5%), while $3\times$ amplification increases SS to 30.2% and reduces anti-stereotype preference to 69.8%, demonstrating more predictable behavior.

### 3.8 Individual Feature Causality Validation

To identify which features exhibit genuine causal relationships, we test each of the top 20 bias and anti-bias features individually under both ablation ($0\times$) and amplification ($2\times$) conditions, measuring changes in stereotype score ($\Delta$SS) and LMS. We classify a feature as causal only if both interventions produce consistent directional effects. Results show 12 out of 20 bias features (60%) meet causality criteria, with $\Delta$SS ranging from 1-4 percentage points. Key causal features include Feature 6980 (layers 9-10, 12-14), Feature 3153 (layers 13-17), Feature 7059 (layers 13-15), Feature 5763 (layers 18-19), and Feature 6088 (layers 29-30). In contrast, 0 out of 20 anti-bias features (0%) demonstrate causal relationships, with most showing minimal effects ($\Delta\text{Anti} \approx -1.57\%$) concentrated exclusively in layer 31. This asymmetry suggests that features correlating with anti-stereotypical preferences may be downstream consequences rather than causal drivers of model behavior.

### 3.9 Causal Validation Through Individual Feature Manipulation

To validate genuine causal relationships, we tested each identified feature individually under both ablation and amplification conditions. A feature is classified as causal only if both interventions produce consistent directional effects on stereotype scores. Figure 3 presents the intervention effects for the 12 features that met causality criteria.

The asymmetry between bias and anti-bias features suggests distinct computational mechanisms: bias features appear to causally drive stereotypical behavior, while anti-bias features may represent downstream consequences of anti-stereotypical processing rather than causal drivers.

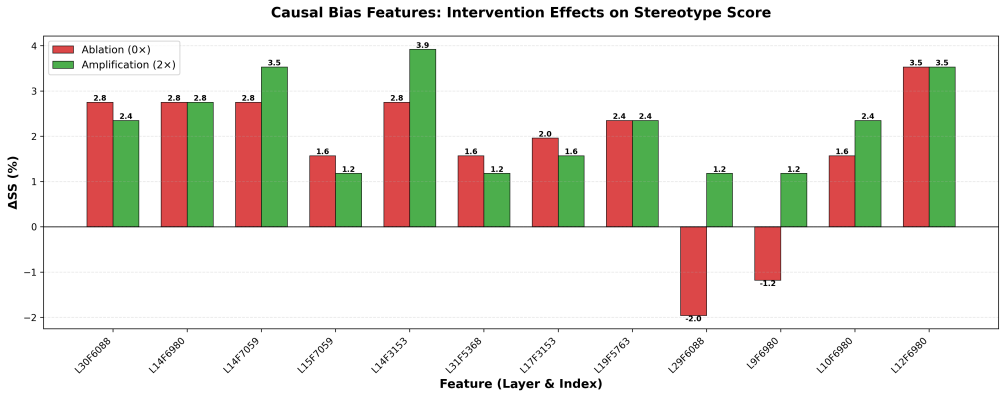


Figure 3: Individual causal feature intervention effects. Each of the 12 validated causal bias features is tested under ablation (0×, red bars) and amplification (2×, green bars). Bars show the change in stereotype score (ΔSS) relative to baseline. Features span layers 9-31, with the strongest effects observed in layer 14 (Features 6980, 7059, 3153) and layer 30 (Feature 6088). Consistent positive ΔSS under both conditions confirms causal relationships: removing or amplifying these features reliably shifts model behavior toward more stereotypical outputs. Out of 20 tested bias features, 12 (60%) demonstrated causality, while 0 out of 20 anti-bias features showed causal effects.

# 4 Proposed Theoretical Method: Increment-SAE with Calibrated Leakage Accounting

**Context: baseline vs. our extension.** Our baseline experiments follow standard representation-editing / feature-surgery evaluations for bias mitigation (e.g., selecting features and intervening at hidden states, then measuring behavioral bias on StereoSet). Our extension turns this into a single end-to-end *mechanism-grounded* and *accounted* procedure: we (i) represent the model's *causal transition mediator* via an Increment-SAE on residual updates, (ii) enforce one-step consistency with the frozen transformer dynamics, (iii) train the encoder as a *worst-case extractor* against an explicit adversary family, and (iv) calibrate a stochastic edit to meet a declared leakage budget under a restricted (hypothesis) DP-style notion.

## 4.1 Objects and residual update mediator

Let $x$ denote a StereoSet prompt with candidate completions. Let $A(x) \in \{0, 1\}$ denote the sensitive attribute indicator (e.g., gender group; or a counterfactual label under swap). Let the residual stream at layer $\ell$ be $H_\ell(x) \in \mathbb{R}^{T\times d}$, with frozen block update map

$$H_{\ell+1}(x) = \Phi_\ell(H_\ell(x)). \tag{4}$$

We define the *exact* discrete residual increment (no PDE approximation):

$$\Delta H_\ell(x) := H_{\ell+1}(x) - H_\ell(x). \tag{5}$$

Our interventions act on $\Delta H_\ell$, which is the mediator of depth evolution under the recursion $H_{\ell+1} = H_\ell + \Delta H_\ell$.

## 4.2 Increment-SAE on residual updates (mediator representation)

We train a sparse autoencoder over the increments $\Delta H_\ell(x)$ (not only on $H_\ell$). For a shared encoder $E$ and layer-specific decoders $D_\ell$, we compute codes

$$Z_\ell(x) = E(\Delta H_\ell(x)) = \mathrm{ReLU}\big(\Delta H_\ell(x) W_E^\top + \mathbf{1} b_E^\top\big) \in \mathbb{R}^{T\times m}, \tag{6}$$

and reconstruct the increment via

$$\widehat{\Delta H}_\ell(x) = D_\ell(Z_\ell(x)) = Z_\ell(x) W_{D,\ell}^\top \in \mathbb{R}^{T\times d}. \tag{7}$$

We interpret $Z_\ell(x)$ as a sparse coordinate system for the *transition mediator* $\Delta H_\ell(x)$.

### 4.3 Mechanism grounding via one-step consistency

To ensure SAE features align with the transformer transition, we enforce a one-step consistency constraint. Define $\widehat{H}_{\ell+1}(x) := H_\ell(x) + \widehat{\Delta H}_\ell(x)$ and penalize mismatch with frozen dynamics:

$$\mathcal{L}_{\text{step}} = \sum_{\ell=1}^{L-1} \mathbb{E}_x \left[ \left\| \widehat{H}_{\ell+1}(x) - \Phi_\ell(H_\ell(x)) \right\|_F^2 \right]. \tag{8}$$

This grounds the SAE in the exact discrete update equation (no continuum assumptions).

### 4.4 Worst-case extractor SAE (explicit adversary class)

We declare an adversary family by defining a joint representation over edited layers. Let $\mathcal{L} = \{\ell_1, \ldots, \ell_k\}$ be selected layers, and $S_\ell \subseteq \{1, \ldots, m\}$ be selected code coordinates. We form a joint code

$$Z_{\text{joint}}(x) := \text{concat}\big(Z_{\ell_1, S_{\ell_1}}(x), \ldots, Z_{\ell_k, S_{\ell_k}}(x)\big), \tag{9}$$

and define the hypothesis family

$$\mathcal{F} := \{f(x) = g(Z_{\text{joint}}(x)) \ : \ g \in \mathcal{G}\}. \tag{10}$$

We use a nested ladder $\mathcal{G} = \{\mathcal{G}_1, \mathcal{G}_2, \mathcal{G}_3\}$ (increasing power), e.g.: $\mathcal{G}_1$ logistic regression; $\mathcal{G}_2$ 2-layer MLP; $\mathcal{G}_3$ deeper MLP / RBF-SVM. To avoid "the encoder missed leakage", we train the SAE encoder to *expose* sensitive information by a bilevel term:

$$\min_{E, \{D_\ell\}} \ \mathcal{L}_{\text{recon}} + \lambda_1 \|Z\|_1 + \lambda_{\text{step}} \mathcal{L}_{\text{step}} \ - \ \beta \max_{g \in \mathcal{G}} \ \mathcal{L}_{\text{clf}}(g(Z_{\text{joint}}(x)), A(x)). \tag{11}$$

Here $\mathcal{L}_{\text{recon}} = \sum_\ell \mathbb{E}_x \|\Delta H_\ell(x) - \widehat{\Delta H}_\ell(x)\|_F^2$.

### 4.5 Restricted indistinguishability and leakage accounting

We define neighbors $x \sim x'$ as counterfactual pairs (e.g., gender-swapped minimal edits, or same prompt with different protected realization). We use a restricted (hypothesis) DP-style indistinguishability over $\mathcal{F}$:

$$\forall f \in \mathcal{F}: \quad \Pr[f(x) = 1] \leq e^\varepsilon \Pr[f(x') = 1] + \delta. \tag{12}$$

Practically, we estimate a restricted advantage

$$\tau_{\text{joint}} := \sup_{g \in \mathcal{G}} \left| \Pr[g(Z_{\text{joint}}(x)) = 1] - \Pr[g(Z_{\text{joint}}(x')) = 1] \right|, \tag{13}$$

and map it to a privacy-like scalar

$$\varepsilon_{\mathcal{F},\text{joint}} := \log \frac{1 + \tau_{\text{joint}}}{1 - \tau_{\text{joint}}}. \tag{14}$$

We report $\tau_{\text{joint}}$ and $\varepsilon_{\mathcal{F},\text{joint}}$ for the ladder $\mathcal{G}_k$.

### 4.6 Causal edit on the mediator code

For each selected layer $\ell \in \mathcal{L}$, we intervene on the mediator coordinates $Z_{\ell,S_\ell}(x)$.

**Deterministic edit (erasure baseline).**

$$Z^{\text{edit}}_{\ell,S_\ell}(x) = \Pi_\ell^\perp \, Z_{\ell,S_\ell}(x), \tag{15}$$

where $\Pi_\ell^\perp$ can be coordinate-zeroing or a learned projection (e.g., nullspace projector).

**Stochastic classifier-calibrated edit (our accounted variant).**

$$Z^{\text{edit}}_{\ell,S_\ell}(x) = \text{clip}(Z_{\ell,S_\ell}(x), C_\ell) + \eta_\ell, \qquad \eta_\ell \sim \mathcal{N}(0, \sigma_\ell^2 I). \tag{16}$$

We decode the edited increment and apply the exact residual step:

$$\widehat{\Delta H}^{\text{edit}}_\ell(x) = D_\ell(Z^{\text{edit}}_\ell(x)), \qquad H^{\text{edit}}_{\ell+1}(x) = H^{\text{edit}}_\ell(x) + \widehat{\Delta H}^{\text{edit}}_\ell(x). \tag{17}$$

Because the intervention modifies $\Delta H_\ell$ (the one-step mediator), its effect propagates causally through depth under the model recursion.

**Algorithm 1: Calibrated causal mediator edit (our extension)**
**Input:** frozen model $\{\Phi_\ell\}$, dataset of pairs $(x, x')$, target $\tau^\star$, layers $\mathcal{L}$, coords $S_\ell$, ladder $\mathcal{G}$.
1) Compute increments $\Delta H_\ell(x) = \Phi_\ell(H_\ell(x)) - H_\ell(x)$.
2) Train Increment-SAE with step consistency + worst-case extractor objective (11).
3) For candidate $\sigma$: edit codes $Z_{\ell,S_\ell} \mapsto \mathrm{clip}(\cdot, C_\ell) + \mathcal{N}(0, \sigma^2 I)$, decode $\widehat{\Delta H}_\ell^{\mathrm{edit}}$, and roll out $H_{\ell+1}^{\mathrm{edit}} = H_\ell^{\mathrm{edit}} + \widehat{\Delta H}_\ell^{\mathrm{edit}}$.
4) Evaluate leakage $\tau_{\mathrm{joint}}(\sigma) = \sup_{g\in\mathcal{G}} |\Pr[g(Z_{\mathrm{joint}}(x))] - \Pr[g(Z_{\mathrm{joint}}(x'))]|$.
5) Binary search for $\sigma^\star$ s.t. $\tau_{\mathrm{joint}}(\sigma^\star) \le \tau^\star$.
**Output:** edited model rollouts, $(\tau_{\mathrm{joint}}, \varepsilon_{\mathcal{F},\mathrm{joint}})$, noise energy $\sum_\ell |S_\ell|\sigma^{\star 2}$, and StereoSet metrics.

### 4.7 End-to-end calibration and two-number accounting

**Leakage calibration.** Given a target leakage budget $\tau^\star$, we select the minimum noise scale satisfying

$$\sigma^\star := \inf\{\sigma : \tau_{\mathrm{joint}}(\sigma) \le \tau^\star\}, \tag{18}$$

found by monotone search (binary search) over $\sigma$.

**Noise energy budget.** For isotropic noise on selected code coordinates,

$$\mathbb{E}\|\eta\|_2^2 = \sum_{\ell\in\mathcal{L}} |S_\ell|\sigma_\ell^2. \tag{19}$$

If $\sigma_\ell \equiv \sigma$ across layers, $\mathbb{E}\|\eta\|_2^2 = \sigma^2 \sum_{\ell\in\mathcal{L}} |S_\ell|$. We report both (i) leakage numbers $(\tau_{\mathrm{joint}}, \varepsilon_{\mathcal{F},\mathrm{joint}})$ and (ii) injected noise energy.

## 5 Future Work

This work shows that a small set of SAE features has a measurable causal effect on stereotypical model behavior and that these features localize to specific middle and late transformer layers. However, interventions are currently applied manually and uniformly across contexts. An open direction is to learn rules or controllers that activate or suppress identified causal features conditionally during inference. This would allow bias mitigation to be applied only when relevant, without modifying model parameters or affecting unrelated behaviors. It also remains to be tested whether repeated feature-level interventions are stable over long generations or interact with decoding strategies such as temperature and nucleus sampling.

Our results also leave several mechanistic questions unresolved. While a majority of stereotypical features exhibit causal effects, anti-stereotypical features do not, and are concentrated almost entirely in the final layer. This suggests that anti-stereotypical behavior may not be driven by independent causal features, but this hypothesis has not been directly tested. Future work should examine whether earlier-layer features indirectly mediate anti-stereotypical outcomes, or whether such behavior emerges from late-stage decoding dynamics. Extending the analysis to other bias attributes, model sizes, and training regimes is necessary to determine which aspects of feature localization and causal asymmetry generalize beyond the specific setting studied here.

## 6 Conclusion

We demonstrate that Sparse Autoencoders with shared encoders can identify interpretable bias features in transformer language models and enable targeted manipulation without retraining. Applied to Llama-3-8B, our method successfully identified features in layers 9-31 that causally influence gender bias, with 60% of bias features exhibiting genuine causal effects under individual testing. Critically, we observe a striking asymmetry: bias features demonstrate clear causal relationships when ablated or amplified, while anti-bias features show minimal causal effects, appearing instead as downstream consequences of model processing. This asymmetry reveals that bias and fairness operate through distinct computational mechanisms, challenging the assumption that bias mitigation simply involves inverting bias pathways. Our feature level intervention approach enables surgical bias

modification while preserving unrelated model capabilities, though unexpected results in collective feature manipulation highlight the complexity of bias encoding and the need for rigorous causal validation before deployment.

## References


Hyojin Bahng, Ali Jahanian, Suha Kwak, et al. Learning de-biased representations with biased representations. ICML 2020; arXiv:1910.02806, 2020.

Nicholas Belrose, Daniel Schneider-Joseph, Shauli Ravfogel, Ryan Cotterell, Edward Raff, et al. Leace: Perfect linear concept erasure in closed form. NeurIPS 2023; arXiv:2306.03819, 2023.

Lucas Bourtoule, Varun Chandrasekaran, Christopher A. Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. Machine unlearning, 2020. URL https://arxiv.org/abs/1912.03817.

Somnath Basu Roy Chowdhury, Nicholas Monath, Avinava Dubey, Amr Ahmed, and Snigdha Chaturvedi. Robust concept erasure via kernelized rate-distortion maximization. NeurIPS 2023; arXiv:2312.00194, 2023.

Hoagy Cunningham, Aidan Ewart, Logan Riggs, Robert Huben, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models, 2023. URL https://arxiv.org/abs/2309.08600.

Praveen Hegde. Effectiveness of sparse autoencoder for understanding and removing gender bias in LLMs. In *NeurIPS 2024 Workshop on Scientific Methods for Understanding Deep Learning*, 2024. URL https://openreview.net/forum?id=IxEMXN5hCg.

Yong Lin, Hangyu Lin, Wei Xiong, Shizhe Diao, Jianmeng Liu, Jipeng Zhang, Rui Pan, Haoxiang Wang, Wenbin Hu, Hanning Zhang, Hanze Dong, Renjie Pi, Han Zhao, Nan Jiang, Heng Ji, Yuan Yao, and Tong Zhang. Mitigating the alignment tax of rlhf, 2024. URL https://arxiv.org/abs/2309.06256.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback, 2022. URL https://arxiv.org/abs/2203.02155.

Arthur Saillenfest and William Lemberger. Nonlinear concept erasure: a density matching approach. arXiv:2507.12341, 2025. Introduces a density-matching nonlinear concept erasure approach (EOPARD/LEOPARD).

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017. URL https://arxiv.org/abs/1707.06347.

Wei Shi, Sihang Li, Tao Liang, Mingyang Wan, Guojun Ma, Xiang Wang, and Xiangnan He. Route sparse autoencoder to interpret large language models, 2025. URL https://arxiv.org/abs/2503.08200.

Shuyi Zhang, Wei Shi, Sihang Li, Jiayi Liao, Tao Liang, Hengxing Cai, and Xiang Wang. Interpretable reward model via sparse autoencoder, 2025. URL https://arxiv.org/abs/2508.08746.